Second-order LOD multigrid method for multidimensional Riesz fractional diffusion equation

Minghua Chen, Yantao Wang, Xiao Cheng, Weihua Deng*

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, P.R. China

Abstract

We propose a locally one dimensional (LOD) finite difference method for multidimensional Riesz fractional diffusion equation with variable coefficients on a finite domain. The numerical method is second-order convergent in both space and time directions, and its unconditional stability is strictly proved. Comparing with the popular first-order finite difference method for fractional operator, the form of obtained matrix algebraic equation is changed from $(I-A)u^{k+1}=u^k+b^{k+1}$ to $(I-\widetilde{A})u^{k+1}=(I+\widetilde{B})u^k+\tilde{b}^{k+1/2}$; the three matrices A, $\widetilde{A}$ and $\widetilde{B}$ are all Toeplitz-like, i.e., they have completely same structure and the computational count for matrix vector multiplication is $\mathcal{O}(N\log N)$; and the computational costs for solving the two matrix algebraic equations are almost the same. The LOD-multigrid method is used to solve the resulting matrix algebraic equation, and the computational count is $\mathcal{O}(N\log N)$ and the required storage is $\mathcal{O}(N)$, where N is the number of grid points. Finally, the extensive numerical experiments are performed to show the powerfulness of the second-order scheme and the LOD-multigrid method.

Keywords: Riesz fractional diffusion equation; Second-order discretization; Toeplitz and Circulant matrices; Multigrid method

Mathematics Subject Classification (2010): 35R11, 65M06, 65M55

1. Introduction

In recent years considerable interests in fractional calculus have been stimulated by the applications in physical, chemical, biological, and engineering, etc., areas [9]. The definitions of fractional calculus are versatile, e.g., Riemann-Liouville derivative, Grünwald-Letnikov derivative, Caputo derivative, Weyl derivative and Riesz derivative et al [11, 14], and they are not completely equivalent. Depending on the particular applied field, sometimes one of its definitions is more popular than others. For example, the Riesz fractional

*Corresponding author. E-mail: dengwh@lzu.edu.cn.

derivative appears in the continuous limit of lattice models with long-range interactions [19]. This paper focuses on the multidimensional Riesz fractional diffusion equations.

Nowadays, the finite difference discretization for space fractional derivatives is experiencing rapid development, including the Riesz fractional derivative; such as, Yang et al numerically study the Riesz space fractional PDEs with two different fractional orders $1 < \alpha \leq 2$ and $0 < \beta < 1$ [24]; Zhuang et al consider a variable-order fractional advection-diffusion equation with a nonlinear source term on a finite domain [26]. In the last two years, for the space fractional derivatives, we notice that two different second-order discretization schemes are developed [17, 20]; even the third-order discrezation scheme is obtained [25] if a compact difference operator is performed on the discrezation scheme given in [20].

Another topic related to effectively solving the equations involving fractional operators is about how to efficiently solve the resulting matrix algebraic equations. The 'unlucky' thing is that the matrix in the matrix algebraic equation is usually full because of the nonlocal properties of the fractional operators, and the 'lucky' thing, as pointed out in [12, 21, 22], is that the matrix has some special structure, i.e., the matrix is Toeplitz-like matrix, and the count of its matrix vector multiplication is $\mathcal{O}(N\log N)$ by using the constructed circulant matrix and fast Fourier transform, and the required storage is $\mathcal{O}(N)$, where N is the number of grid points. Pang and Sun [12] successfully use the multigrid method (MGM) to efficiently solve the resulting matrix algebraic equation of the one dimensional fractional diffusion equation by using first-order discretization scheme [10]. Here we further extend the MGM to solve the matrix algebraic equation of the multidimensional Riesz fractional diffusion equation discretized by the second-order scheme.

We use the LOD strategy to solve the multidimensional Riesz fractional diffusion equation. The LOD methods include alternating direction (AD) methods and fractional step procedures [6]. The AD methods were first introduced in three papers [4, 7, 13] by Douglas, Peaceman, and Rachford. The Peaceman and Rachford (PR-AD) method works well for two-dimensional problems. However, it can not be extended to higher dimensional problems. Douglas (D-AD) method [4, 5, 6] are valid for any dimensional equations. And PR-AD and D-AD are equivalent in two-dimensional problems. Both PR-AD and D-AD schemes are used in this paper to discretize the multidimensional Riesz fractional diffusion equation. In each dimension, the obtained matrix algebraic equation is solved by MGM. Although the spacial fractional derivative is discretized by second-order scheme, for any single dimension the form of the obtained matrix algebraic equation is $(I - \widetilde{A})u^{k+1} = (I + \widetilde{B})u^k + \tilde{b}^{k+1/2}$, in fact the corresponding form for the first-order

discretization scheme is $(I-A)u^{k+1} = u^k + b^{k+1}$; the three matrices A, $\widetilde{A}$ and $\widetilde{B}$ are all Toeplitz-like, and they have completely same structure and the computational count for matrix vector multiplication is $\mathcal{O}(N\log N)$; and the computational costs for solving the two matrix algebraic equations are almost the same. In other words, the second-order scheme improves the accuracy but almost without increasing the computational cost.

More concretely, in this paper using the second-order accurate and unconditionally stable computational scheme and LOD-MGM, we solve the following multidimensional variable coefficients Riesz fractional diffusion equation with the computational count $\mathcal{O}(N\log N)$ and storage $\mathcal{O}(N)$,

$$\begin{cases} \dfrac{\partial u(x,y,z,t)}{\partial t} = c(x,y,z,t)\dfrac{\partial^{\alpha} u(x,y,z,t)}{\partial |x|^{\alpha}} + d(x,y,z,t)\dfrac{\partial^{\beta} u(x,y,z,t)}{\partial |y|^{\beta}} \\ \qquad\qquad + e(x,y,z,t)\dfrac{\partial^{\gamma} u(x,y,z,t)}{\partial |z|^{\gamma}} + f(x,y,z,t), \\ u(x,y,z,0) = u_0(x,y,z) \quad \text{for} \quad (x,y,z) \in \Omega, \\ u(x,y,z,t) = 0 \quad \text{for} \quad (x,y,z,t) \in \partial\Omega \times (0,T], \end{cases} \tag{1.1}$$

in the domain $\Omega = (x_L, x_R) \times (y_L, y_R) \times (z_L, z_R)$, $0 < t \le T$, where the orders of the Riesz fractional derivatives are $1 < \alpha, \beta, \gamma < 2$; $f(x,y,z,t)$ is a source term and the variable coefficients $c(x,y,z,t) \ge 0$, $d(x,y,z,t) \ge 0$, $e(x,y,z,t) \ge 0$; the Riesz fractional derivative for $n \in \mathbb{N}$, $n-1 < \nu \le n$, is defined as [3, 19]

$$\frac{\partial^{\nu} u(x,y,z,t)}{\partial |x|^{\nu}} = -\kappa_{\nu} \left({}_{x_L}D_x^{\nu} + {}_xD_{x_R}^{\nu} \right) u(x,y,z,t), \tag{1.2}$$

where the coefficient $\kappa_{\nu} = \frac{1}{2cos(\nu\pi/2)}$, and

$${}_{x_L}D_x^{\nu} u(x,y,z,t) = \frac{1}{\Gamma(n-\nu)} \frac{\partial^n}{\partial x^n} \int_{x_L}^{x} (x-\xi)^{n-\nu-1} u(\xi,y,z,t) d\xi, \tag{1.3}$$

$${}_xD_{x_R}^{\nu} u(x,y,z,t) = \frac{(-1)^n}{\Gamma(n-\nu)} \frac{\partial^n}{\partial x^n} \int_{x}^{x_R} (\xi-x)^{n-\nu-1} u(\xi,y,z,t) d\xi, \tag{1.4}$$

are the left and right Riemann-Liouville space fractional derivatives, respectively.

The outline of this paper is as follows. In the next section, we introduce the second-order finite difference discretizations for the Riesz fractional derivatives; and the full discretization of (1.1) is derived, where the Crank-Nicolson scheme and LOD method are combined together. We theoretically prove the presented finite difference scheme is unconditionally stable in Section 3. In Section 4 we propose a V-cycle LOD-MGM for the resulting system of (1.1). To show the powerfulness of the second-order scheme and LOD-MGM, the extensive numerical experiments are performed in Section 5. Finally, we conclude the paper with some remarks in the last section.

2. Derivation of the finite difference scheme

In this section, we derive the full discretization schemes of (1.1). The first subsection introduces the second-order finite difference discretizations for the Riesz fractional derivatives in a finite domain. Then in the second subsection, we present the scheme for the one dimensional case of (1.1). The third and fourth subsections detailedly provide the two dimensional case of (1.1) and (1.1) itself, respectively.

2.1. Discretizations for the Riesz fractional derivatives

Take the mesh points $x_i = x_L + i\Delta x, i = 0, 1, \ldots, N_x$, $y_j = y_L + j\Delta y, j = 0, 1, \ldots, N_y$, $z_l = z_L + l\Delta z, l = 0, 1, \ldots, N_z$ and $t_k = k\Delta t, k = 0, 1, \ldots, N_t$, where $\Delta x = (x_R - x_L)/N_x$, $\Delta y = (y_R - y_L)/N_y$, $\Delta z = (z_R - z_L)/N_z$, $\Delta t = T/N_t$, i.e., Δx, Δy and Δz are the uniform space stepsizes in the corresponding directions, Δt the time stepsize. For $\nu \in (1, 2)$, the left and right Riemann-Liouville space fractional derivatives (1.3) and (1.4) have the second-order approximation operators $\delta_{\nu,+x} u^k_{i,j,l}$ and $\delta_{\nu,-x} u^k_{i,j,l}$, respectively, given in a finite domain [3, 17], where $u^k_{i,j,l}$ denotes the approximated value of $u(x_i, y_j, z_l, t_k)$.

The approximation operator of (1.3) is defined by [3, 17]

$$\delta_{\nu,+x} u^k_{i,j,l} := \frac{1}{\Gamma(4-\nu)(\Delta x)^\nu} \sum_{m=0}^{i+1} g^\nu_m u^k_{i-m+1,j,l}, \tag{2.1}$$

and there exists

$${}_{x_L}D^\nu_x u(x, y, z, t) = \delta_{\nu,+x} u^k_{i,j,l} + \mathcal{O}(\Delta x)^2, \tag{2.2}$$

where

$$g^\nu_m = \begin{cases} 1, & m = 0, \\ -4 + 2^{3-\nu}, & m = 1, \\ 6 - 2^{5-\nu} + 3^{3-\nu}, & m = 2, \\ (m+1)^{3-\nu} - 4m^{3-\nu} + 6(m-1)^{3-\nu} & \\ \qquad - 4(m-2)^{3-\nu} + (m-3)^{3-\nu}, & m \geq 3. \end{cases} \tag{2.3}$$

Analogously, the approximation operator of (1.4) is described as [3]

$$\delta_{\nu,-x} u^k_{i,j,l} := \frac{1}{\Gamma(4-\nu)(\Delta x)^\nu} \sum_{m=0}^{N_x-i+1} g^\nu_m u^k_{i+m-1,j,l}, \tag{2.4}$$

and it holds that

$${}_{x}D^\nu_{x_R} u(x, y, z, t) = \delta_{\nu,-x} u^k_{i,j,l} + \mathcal{O}(\Delta x)^2, \tag{2.5}$$

where g^ν_m is defined by (2.3).

Combining (2.2) and (2.5), we obtain the approximation operator of the (Riemann-Liouville) Riesz fractional derivative

$$
\begin{aligned}
\left.\frac{\partial^\nu u(x,y_j,z_l,t_k)}{\partial|x|^\nu}\right|_{x=x_i} &= -\kappa_\nu \left({}_{x_L}D_x^\nu + {}_x D_{x_R}^\nu\right) u(x,y_j,z_l,t_k)\big|_{x=x_i} \\
&= -\kappa_\nu \left(\delta_{\nu,+x} + \delta_{\nu,-x}\right) u_{i,j,l}^k + \mathcal{O}(\Delta x)^2 \\
&= \frac{-\kappa_\nu}{\Gamma(4-\nu)\Delta x^\nu}\left(\sum_{m=0}^{i+1} g_m^\nu u_{i-m+1,j,l}^k + \sum_{m=0}^{N_x-i+1} g_m^\nu u_{i+m-1,j,l}^k\right) + \mathcal{O}(\Delta x)^2 \\
&= \frac{-\kappa_\nu}{\Gamma(4-\nu)\Delta x^\nu}\left(\sum_{m=0}^{i+1} g_{i-m+1}^\nu u_{m,j,l}^k + \sum_{m=i-1}^{N_x} g_{m-i+1}^\nu u_{m,j,l}^k\right) + \mathcal{O}(\Delta x)^2 \\
&:= \frac{-\kappa_\nu}{\Gamma(4-\nu)\Delta x^\nu}\sum_{m=0}^{N_x} \widetilde{g}_{i,m}^\nu u_{m,j,l}^k + \mathcal{O}(\Delta x)^2,
\end{aligned}
\tag{2.6}
$$

where

$$
\widetilde{g}_{i,m}^\nu = \begin{cases} g_{i-m+1}^\nu, & m < i-1, \\ g_0^\nu + g_2^\nu, & m = i-1, \\ 2g_1^\nu, & m = i, \\ g_0^\nu + g_2^\nu, & m = i+1, \\ g_{m-i+1}^\nu, & m > i+1, \end{cases} \tag{2.7}
$$

with $i = 1, \ldots, N_x - 1$, together with the Dirichlet boundary conditions that define $u_{0,j,l}^k$ and $u_{N_x,j,l}^k$ as appropriate.

Taking $\nu = 2$, both Eq. (2.2) and (2.5) reduce to the following form

$$
\frac{\partial^2 u(x_i,y,z,t)}{\partial x^2} = \frac{u(x_{i+1},y,z,t) - 2u(x_i,y,z,t) + u(x_{i-1},y,z,t)}{(\Delta x)^2} + \mathcal{O}(\Delta x)^2.
$$

Similarly, it is easy to get the one-dimensional and two-dimensioanl case of (2.1)-(2.7).

2.2. Numerical scheme for 1D

Consider the one-dimensional Riesz fractional diffusion equation

$$
\frac{\partial u(x,t)}{\partial t} = c(x,t)\frac{\partial^\alpha u(x,t)}{\partial|x|^\alpha} + f(x,t). \tag{2.8}
$$

In the time direction, we use the Crank-Nicolson scheme. Taking the uniform time step Δt and space step Δx, and setting $c_i^k = c(x_i,t_k)$ and $f_i^{k+1/2} = f(x_i,t_{k+1/2})$, where $t_{k+1/2} = (t_k + t_{k+1})/2$, the full discretization of (2.8) has the following form

$$
\frac{u_i^{k+1} - u_i^k}{\Delta t} = \frac{-\kappa_\alpha c_i^{k+1/2}}{\Gamma(4-\alpha)\Delta x^\alpha}\sum_{m=0}^{N_x} \widetilde{g}_{i,m}^\alpha \frac{u_m^k + u_m^{k+1}}{2} + f_i^{k+1/2}. \tag{2.9}
$$

Then (2.9) can be expressed as

$$\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}\right)u_i^{k+1}=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}\right)u_i^k+\Delta t f_i^{k+1/2}, \tag{2.10}$$

where

$$\delta''_{\alpha,x}u_i^k:=\frac{-\kappa_\alpha c_i^{k+1/2}}{\Gamma(4-\alpha)\Delta x^\alpha}\sum_{m=0}^{N_x}\widetilde{g}_{i,m}^\alpha u_m^k;\quad \delta''_{\alpha,x}u_i^{k+1}:=\frac{-\kappa_\alpha c_i^{k+1/2}}{\Gamma(4-\alpha)\Delta x^\alpha}\sum_{m=0}^{N_x}\widetilde{g}_{i,m}^\alpha u_m^{k+1}.$$

Putting $\xi_i^{k+1/2}=\frac{-\Delta t\kappa_\alpha c_i^{k+1/2}}{2\Gamma(4-\alpha)\Delta x^\alpha}$, the system of equations given by (2.10) takes the form

$$(I-A^{k+1/2})U^{k+1}=(I+A^{k+1/2})U^k+\Delta t F^{k+1/2}, \tag{2.11}$$

where I is the $(N_x-1)\times(N_x-1)$ identity matrix,

$$U^k=[u_1^k,u_2^k,\ldots,u_{N_x-1}^k]^{\mathrm{T}},\quad F^{k+1/2}=[f_1^{k+1/2},f_2^{k+1/2},\ldots,f_{N_x-1}^{k+1/2}]^{\mathrm{T}},$$

and the discretizations at the interior x-gridpoints define the entries of the matrix $A^{k+1/2}$, $A_{i,m}^{k+1/2}$ for $i=1,\ldots,N_x-1$ and $m=1,\ldots,N_x-1$ are defined by

$$A_{i,m}^{k+1/2}=\begin{cases} g_{i-m+1}^\alpha\xi_i^{k+1/2}, & m<i-1,\\ (g_0^\alpha+g_2^\alpha)\xi_i^{k+1/2}, & m=i-1,\\ 2g_1^\alpha\xi_i^{k+1/2}, & m=i,\\ (g_0^\alpha+g_2^\alpha)\xi_i^{k+1/2}, & m=i+1,\\ g_{m-i+1}^\alpha\xi_i^{k+1/2}, & m>i+1.\end{cases} \tag{2.12}$$

2.3. LOD scheme for 2D

Consider the following two-dimensional Riesz fractional diffusion equation

$$\frac{\partial u(x,y,t)}{\partial t}=c(x,y,t)\frac{\partial^\alpha u(x,y,t)}{\partial|x|^\alpha}+d(x,y,t)\frac{\partial^\beta u(x,y,t)}{\partial|y|^\beta}+f(x,y,t). \tag{2.13}$$

Analogously we still use the Crank-Nicolson scheme to do the discretization in time direction. Taking $u_{i,j}^k$ as the approximated value of $u(x_i,y_j,t_k)$, $c_{i,j}^k=c(x_i,y_j,t_k)$, $d_{i,j}^k=d(x_i,y_j,t_k)$, $t_{n+1/2}=(t_n+t_{n+1})/2$, $f_{i,j}^{k+1/2}=f(x_i,y_j,t_{k+1/2})$, $\Delta x=(x_R-x_L)/N_x$, and $\Delta y=(y_R-y_L)/N_y$, for the uniform space steps $\Delta x,\Delta y$ and the time stepsize Δt, the resulting discretization of (2.13) can be written as

$$\begin{aligned}\frac{u_{i,j}^{k+1}-u_{i,j}^k}{\Delta t}=&\frac{-\kappa_\alpha c_{i,j}^{k+1/2}}{\Gamma(4-\alpha)\Delta x^\alpha}\sum_{m=0}^{N_x}\widetilde{g}_{i,m}^\alpha\frac{u_{m,j}^k+u_{m,j}^{k+1}}{2}\\ &+\frac{-\kappa_\beta d_{i,j}^{k+1/2}}{\Gamma(4-\beta)\Delta x^\beta}\sum_{m=0}^{N_y}\widetilde{g}_{j,m}^\beta\frac{u_{i,m}^k+u_{i,m}^{k+1}}{2}+f_{i,j}^{k+1/2}.\end{aligned} \tag{2.14}$$

Similarly, we define

$$\delta''_{\alpha,x}u^k_{i,j}=\frac{-\kappa_\alpha c^{k+1/2}_{i,j}}{\Gamma(4-\alpha)\Delta x^\alpha}\sum_{m=0}^{N_x}\widetilde{g}^\alpha_{i,m}u^k_{m,j};\quad \delta''_{\alpha,x}u^{k+1}_{i,j}=\frac{-\kappa_\alpha c^{k+1/2}_{i,j}}{\Gamma(4-\alpha)\Delta x^\alpha}\sum_{m=0}^{N_x}\widetilde{g}^\alpha_{i,m}u^{k+1}_{m,j};$$

$$\delta''_{\beta,y}u^k_{i,j}=\frac{-\kappa_\beta d^{k+1/2}_{i,j}}{\Gamma(4-\beta)\Delta y^\beta}\sum_{m=0}^{N_y}\widetilde{g}^\beta_{j,m}u^k_{i,m};\quad \delta''_{\beta,y}u^{k+1}_{i,j}=\frac{-\kappa_\beta d^{k+1/2}_{i,j}}{\Gamma(4-\beta)\Delta y^\beta}\sum_{m=0}^{N_y}\widetilde{g}^\beta_{j,m}u^{k+1}_{i,m};$$

then Eq. (2.14) can be rewritten as

$$\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}-\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^{k+1}_{i,j}=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}+\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^k_{i,j}+\Delta t f^{k+1/2}_{i,j}. \tag{2.15}$$

For the two-dimensional Riesz fractional diffusion equation (2.13), the relevant perturbation of (2.15) is of the form

$$\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}\right)\left(1-\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^{k+1}_{i,j}=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}\right)\left(1+\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^k_{i,j}+\Delta t f^{k+1/2}_{i,j}. \tag{2.16}$$

The scheme (2.16) differs from (2.15) by a perturbation [18]

$$\frac{(\Delta t)^2}{4}\delta''_{\alpha,x}\delta''_{\beta,y}(u^{k+1}_{i,j}-u^k_{i,j}),$$

which may be deduced by distributing the operator products in (2.16). Since $(u^{k+1}_{i,j}-u^k_{i,j})$ is an $\mathcal{O}(\Delta t)$ term, it follows that the perturbation contributes an $\mathcal{O}((\Delta t)^2)$ error component to the truncation error of (2.15). Thus, the scheme (2.16) has a truncation error also $\mathcal{O}((\Delta x)^2)+\mathcal{O}((\Delta y)^2)+\mathcal{O}((\Delta t)^2)$.

For efficiently solving system (2.16), the following techniques can be used:

D-AD scheme [4, 6]:

$$\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}\right)u^*_{i,j}=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}+\Delta t\delta''_{\beta,y}\right)u^k_{i,j}+\Delta t f^{k+1/2}_{i,j}; \tag{2.17}$$

$$\left(1-\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^{k+1}_{i,j}=u^*_{i,j}-\frac{\Delta t}{2}\delta''_{\beta,y}u^k_{i,j}; \tag{2.18}$$

where $u^*_{i,j}$ is an intermediate solution. Subtracting (2.18) from (2.17), we obtain

$$u^*_{i,j}=u^{k+1}_{i,j}+\frac{\Delta t}{2}\delta''_{\beta,y}\left(u^k_{i,j}-u^{k+1}_{i,j}\right).$$

PR-AD scheme [13]:

$$\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}\right)u^*_{i,j}=\left(1+\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^k_{i,j}+\frac{\Delta t}{2}f^{k+1/2}_{i,j}; \tag{2.19}$$

$$\left(1-\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^{k+1}_{i,j}=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}\right)u^*_{i,j}+\frac{\Delta t}{2}f^{k+1/2}_{i,j}; \tag{2.20}$$

with intermediate solution $u^*_{i,j}$. Subtracting (2.20) from (2.19), we have

$$2u^*_{i,j}=\left(1-\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^{k+1}_{i,j}+\left(1+\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^k_{i,j}.$$

2.4. D-AD scheme for 3D

Similarly, the resulting discretization of (1.1) can be written as,

$$\begin{aligned}&\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}-\frac{\Delta t}{2}\delta''_{\beta,y}-\frac{\Delta t}{2}\delta''_{\gamma,z}\right)u^{k+1}_{i,j,l}\\&\quad=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}+\frac{\Delta t}{2}\delta''_{\beta,y}+\frac{\Delta t}{2}\delta''_{\gamma,z}\right)u^{k}_{i,j,l}+f^{k+1/2}_{i,j,l}\Delta t.\end{aligned}\tag{2.21}$$

The perturbation equation of (2.21) is of the form

$$\begin{aligned}&\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}\right)\left(1-\frac{\Delta t}{2}\delta''_{\beta,y}\right)\left(1-\frac{\Delta t}{2}\delta''_{\gamma,z}\right)u^{k+1}_{i,j,l}\\&\quad=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}\right)\left(1+\frac{\Delta t}{2}\delta''_{\beta,y}\right)\left(1+\frac{\Delta t}{2}\delta''_{\gamma,z}\right)u^{k}_{i,j,l}+f^{k+1/2}_{i,j,l}\Delta t.\end{aligned}\tag{2.22}$$

The scheme (2.22) differs from (2.21) by the perturbation term

$$\frac{(\Delta t)^2}{4}(\delta''_{\alpha,x}\delta''_{\beta,y}+\delta''_{\alpha,x}\delta''_{\gamma,z}+\delta''_{\beta,y}\delta''_{\gamma,z})(u^{k+1}_{i,j,l}-u^{k}_{i,j,l})-\frac{(\Delta t)^3}{8}\delta''_{\alpha,x}\delta''_{\beta,y}\delta''_{\gamma,z}(u^{k+1}_{i,j,l}-u^{k}_{i,j,l}).$$

The system of the equations defined by (2.22) can be solved by the D-AD scheme [5, 6]

$$\left(1-\frac{\Delta t}{2}\delta''_{\alpha,x}\right)u^{k,1}_{i,j,l}=\left(1+\frac{\Delta t}{2}\delta''_{\alpha,x}+\Delta t\delta''_{\beta,y}+\Delta t\delta''_{\gamma,z}\right)u^{k}_{i,j,l}+\Delta t f^{k+1/2}_{i,j,l};\tag{2.23}$$

$$\left(1-\frac{\Delta t}{2}\delta''_{\beta,y}\right)u^{k,2}_{i,j,l}=u^{k,1}_{i,j,l}-\frac{\Delta t}{2}\delta''_{\beta,y}u^{k}_{i,j,l};\tag{2.24}$$

$$\left(1-\frac{\Delta t}{2}\delta''_{\gamma,z}\right)u^{k+1}_{i,j,l}=u^{k,2}_{i,j,l}-\frac{\Delta t}{2}\delta''_{\gamma,z}u^{k}_{i,j,l}.\tag{2.25}$$

For maintaining the consistency, we need to carefully specify the boundary conditions of $u^{n,1}_{i,j,l}$ and $u^{k,2}_{i,j,l}$. According to (2.23)-(2.25), we obtain

$$\begin{aligned}&u^{k,1}_{i,j,l}=u^{k+1}_{i,j,l}+\frac{\Delta t}{2}\left(\delta''_{\beta,y}+\delta''_{\gamma,z}\right)\left(u^{k}_{i,j,l}-u^{k+1}_{i,j,l}\right)+\frac{(\Delta t)^2}{4}\delta''_{\beta,y}\delta''_{\gamma,z}u^{k+1}_{i,j,l};\\&u^{k,2}_{i,j,l}=u^{k+1}_{i,j,l}+\frac{\Delta t}{2}\delta''_{\gamma,z}\left(u^{k}_{i,j,l}-u^{k+1}_{i,j,l}\right).\end{aligned}$$

3. Convergence and Stability Analysis

We show the convergence for one-dimensional and multidimensional Riesz fractional diffusion equation by proving the consistency and stability (according to Lax's equivalence theorem).

Lemma 3.1 ([3]). *The coefficients $\widetilde{g}^{\nu}_{i,m}$, $\nu\in(1,2)$ defined in (2.7) satisfy*

$$\begin{aligned}&(1)\ \ \widetilde{g}^{\nu}_{i,i}<0,\quad \widetilde{g}^{\nu}_{i,m}>0\ (m\neq i);\\&(2)\ \ \sum_{m=0}^{N_x}\widetilde{g}^{\nu}_{i,m}<0\ \text{ and }\ -\widetilde{g}^{\nu}_{i,i}>\sum_{m=0,m\neq i}^{N_x}\widetilde{g}^{\nu}_{i,m}.\end{aligned}$$

3.1. The stability of the numerical methods in 1D

Theorem 3.2. The Crank-Nicholson scheme (2.11) of the Riesz fractional diffusion equation (2.9) with $1 < \alpha < 2$ is unconditionally stable.

Proof. First, we prove that the eigenvalues of the matrix $A^{k+1/2}$ have negative real parts. Note that $A_{i,i}^{k+1/2} = \widetilde{g}_{i,i}^{\alpha} \xi_i^{k+1/2}$, and from Lemma 3.1 we obtain

$$r_i = \sum_{m=0, m\neq i}^{N_x} |A_{i,m}^{k+1/2}| = \xi_i^{k+1/2} \sum_{m=0, m\neq i}^{N_x} \widetilde{g}_{i,m}^{\alpha} < -A_{i,i}^{k+1/2}. \tag{3.1}$$

According to the Greschgorin theorem [8], the eigenvalues of the matrix $A^{k+1/2}$ are in the disks centered at $A_{i,i}^{k+1/2}$, with radius r_i, i.e., the eigenvalues λ of the matrix $A^{k+1/2}$ satisfy

$$|\lambda - A_{i,i}^{k+1/2}| \leq r_i, \tag{3.2}$$

thus, the eigenvalues of the matrix $A^{k+1/2}$ have negative real parts. Similarly, we can prove that the eigenvalues of the matrix $I - A^{k+1/2}$ have a magnitude greater than 1 and invertible.

Note that λ is an eigenvalue of the matrix $A^{k+1/2}$ if and only if $1 - \lambda$ is an eigenvalue of the matrix $I - A^{k+1/2}$, if and only if $(1-\lambda)^{-1}(1+\lambda)$ is an eigenvalue of the matrix $(I - A^{k+1/2})^{-1}(I + A^{k+1/2})$. Since $\Re(\lambda) < 0$, it implies that $|(1-\lambda)^{-1}(1+\lambda)| < 1$. Hence, the spectral radius of the matrix $(I - A^{k+1/2})^{-1}(I + A^{k+1/2})$ is less than 1. □

3.2. The stability of the numerical methods in 2D

Under a commutativity assumption for the operators $\left(1 - \frac{\Delta t}{2}\delta''_{\alpha,x}\right)$ and $\left(1 - \frac{\Delta t}{2}\delta''_{\beta,y}\right)$ in (2.15), the PR-AD scheme and D-AD scheme will be shown to be unconditionally stable. The commutativity assumption for these two operators is a common practice in establishing stability of the classical AD methods for the diffusion [6, 18]. The commutativity of these operators implies that the matrices $A_{2D,\Delta x}^{k+1/2}$ and $A_{2D,\Delta y}^{k+1/2}$ given in (3.7) commute.

Theorem 3.3. Both the D-AD scheme (2.17)-(2.18) and PR-AD scheme (2.19)-(2.20), defined by (2.16), are unconditionally stable for $\alpha, \beta \in (1, 2)$, if the matrices $A_{2D,\Delta x}^{k+1/2}$ and $A_{2D,\Delta y}^{k+1/2}$ commute.

Proof. D-AD scheme (2.17)-(2.18) can be expressed in the form

$$(I - A_{2D,\Delta x}^{k+1/2})U^* = (I + A_{2D,\Delta x}^{k+1/2} + 2A_{2D,\Delta y}^{k+1/2})U^k + \Delta t F^{k+1/2}; \tag{3.3}$$

$$(I - A_{2D,\Delta y}^{k+1/2})U^{k+1} = U^* - A_{2D,\Delta y}^{k+1/2}U^k; \tag{3.4}$$

and PR-AD scheme (2.19)-(2.20) is of the form

$$(I - A_{2D,\Delta x}^{k+1/2})U^* = (I + A_{2D,\Delta y}^{k+1/2})U^k + \frac{\Delta t}{2}F^{k+1/2}; \tag{3.5}$$

$$(I - A_{2D,\Delta y}^{k+1/2})U^{k+1} = (I + A_{2D,\Delta x}^{k+1/2})U^{*} + \frac{\Delta t}{2}F^{k+1/2}; \tag{3.6}$$

where the matrices $A_{2D,\Delta x}^{k+1/2}$ and $A_{2D,\Delta y}^{k+1/2}$ denote the operators $\frac{\Delta t}{2}\delta''_{\alpha,x}$ and $\frac{\Delta t}{2}\delta''_{\beta,y}$, and

$$U^k = [u_{1,1}^k, u_{2,1}^k, \ldots, u_{N_x-1,1}^k, u_{1,2}^k, u_{2,2}^k, \ldots, u_{N_x-1,2}^k, \ldots, u_{1,N_y-1}^k, u_{2,N_y-1}^k, \ldots, u_{N_x-1,N_y-1}^k]^T,$$
$$U^* = [u_{1,1}^*, u_{2,1}^*, \ldots, u_{N_x-1,1}^*, u_{1,2}^*, u_{2,2}^*, \ldots, u_{N_x-1,2}^*, \ldots, u_{1,N_y-1}^*, u_{2,N_y-1}^*, \ldots, u_{N_x-1,N_y-1}^*]^T,$$

and the vector $F^{k+1/2}$ absorbs the source terms $f_{i,j}^{k+1/2}$ and the Dirichlet boundary conditions at time $t = t_{k+1}$ in the discretized equation. The matrices $A_{2D,\Delta x}^{k+1/2}$ and $A_{2D,\Delta y}^{k+1/2}$ are matrices of size $(N_x - 1)(N_y - 1) \times (N_x - 1)(N_y - 1)$.

Let us cancel the intermediate solution U^*, then D-AD scheme and PR-AD scheme have the same form

$$(I - A_{2D,\Delta x}^{k+1/2})(I - A_{2D,\Delta y}^{k+1/2})U^{k+1} = (I + A_{2D,\Delta x}^{k+1/2})(I + A_{2D,\Delta y}^{k+1/2})U^{k} + \Delta t F^{k+1/2}, \tag{3.7}$$

from (3.7) we have the perturbation equation

$$(I - A_{2D,\Delta x}^{k+1/2})(I - A_{2D,\Delta y}^{k+1/2})E^{k+1} = (I + A_{2D,\Delta x}^{k+1/2})(I + A_{2D,\Delta y}^{k+1/2})E^{k},$$

where

$$E^k = [e_{1,1}^k, e_{2,1}^k, \ldots, e_{N_x-1,1}^k, e_{1,2}^k, e_{2,2}^k, \ldots, e_{N_x-1,2}^k, \ldots, e_{1,N_y-1}^k, e_{2,N_y-1}^k, \ldots, e_{N_x-1,N_y-1}^k]^T,$$

and $e_{i,j}^k = u(x_i, y_j, t_k) - u_{i,j}^k$, consequently

$$E^k = [(I - A_{2D,\Delta y}^{k+1/2})^{-1}(I - A_{2D,\Delta x}^{k+1/2})^{-1}(I + A_{2D,\Delta x}^{k+1/2})(I + A_{2D,\Delta y}^{k+1/2})]^k E^0.$$

Since the matrices $A_{2D,\Delta x}^{k+1/2}$ and $A_{2D,\Delta y}^{k+1/2}$ commute, it can be written as

$$E^k = [(I - A_{2D,\Delta x}^{k+1/2})^{-1}(I + A_{2D,\Delta x}^{k+1/2})]^k[(I - A_{2D,\Delta y}^{k+1/2})^{-1}(I + A_{2D,\Delta y}^{k+1/2})]^k E^0.$$

According to Theorem 3.2, similarly, it is easy to check that the eigenvalues of the matrix $A_{2D,\Delta x}^{k+1/2}$ and $A_{2D,\Delta y}^{k+1/2}$ have negative real parts. Then, both the spectral radius of the matrixes $(I - A_{2D,\Delta x}^{k+1/2})^{-1}(I + A_{2D,\Delta x}^{k+1/2})$ and $(I - A_{2D,\Delta y}^{k+1/2})^{-1}(I + A_{2D,\Delta y}^{k+1/2})$ are less than 1, therefore the sequence $[(I - A_{2D,\Delta x}^{k+1/2})^{-1}(I + A_{2D,\Delta x}^{k+1/2})]^k$ and $[(I - A_{2D,\Delta y}^{k+1/2})^{-1}(I + A_{2D,\Delta y}^{k+1/2})]^k$ converge to zero matrix [16]. Hence, the difference scheme (2.16) is unconditionally stable. $\square$

3.3. The stability of the numerical methods in 3D

Theorem 3.4. The D-AD scheme (2.23)-(2.25), defined by (2.22), is unconditionally stable for $\alpha, \beta, \gamma \in (1, 2)$, if the matrices $A_{3D,\Delta x}^{k+1/2}$, $A_{3D,\Delta y}^{k+1/2}$ and $A_{3D,\Delta z}^{k+1/2}$ commute.

Proof. D-AD scheme (2.23)-(2.25) can be written as

$$(I - A_{3D,\Delta x}^{k+1/2})U^{k,1} = (I + A_{3D,\Delta x}^{k+1/2} + 2A_{3D,\Delta y}^{k+1/2} + 2A_{3D,\Delta z}^{k+1/2})U^k + \Delta t F^{k+1/2}; \tag{3.8}$$

$$(I - A_{3D,\Delta y}^{k+1/2})U^{k,2} = U^{k,1} - A_{3D,\Delta y}^{k+1/2}U^k; \tag{3.9}$$

$$(I - A_{3D,\Delta z}^{k+1/2})U^{k+1} = U^{k,2} - A_{3D,\Delta z}^{k+1/2}U^k; \tag{3.10}$$

according to (3.8)-(3.10), we have the following equation

$$\begin{aligned}(I - A_{3D,\Delta x}^{k+1/2})(I - A_{3D,\Delta y}^{k+1/2})(I - A_{3D,\Delta z}^{k+1/2})U^{k+1} \\ = (I + A_{3D,\Delta x}^{k+1/2})(I + A_{3D,\Delta y}^{k+1/2})(I + A_{3D,\Delta z}^{k+1/2})U^k + \Delta t F^{k+1/2},\end{aligned} \tag{3.11}$$

where the matrices $A_{3D,\Delta x}^{k+1/2}$ and $A_{3D,\Delta y}^{k+1/2}$ and $A_{3D,\Delta z}^{k+1/2}$ denote the operators $\frac{\Delta t}{2}\delta''_{\alpha,x}$ and $\frac{\Delta t}{2}\delta''_{\beta,y}$ and $\frac{\Delta t}{2}\delta''_{\gamma,z}$, respectively, the vector $U^{k,1}$ and $U^{k,2}$ denote the intermediate solution, the vector $F^{k+1/2}$ absorbs the source terms $f_{i,j,l}^{k+1/2}$ and the Dirichlet boundary conditions at time $t = t_{k+1}$ in the discretized equation. The matrices $A_{3D,\Delta x}^{k+1/2}$ and $A_{3D,\Delta y}^{k+1/2}$ and $A_{3D,\Delta z}^{k+1/2}$ are matrices of size $(N_x - 1)(N_y - 1)(N_z - 1) \times (N_x - 1)(N_y - 1)(N_z - 1)$.

By the similar analysis, it can be proven that the spectral radius of the matrices $(I - A_{3D,\Delta x}^{k+1/2})^{-1}(I + A_{3D,\Delta x}^{k+1/2})$, $(I - A_{3D,\Delta y}^{k+1/2})^{-1}(I + A_{3D,\Delta y}^{k+1/2})$ and $(I - A_{3D,\Delta z}^{k+1/2})^{-1}(I + A_{3D,\Delta z}^{k+1/2})$ are less than 1, therefore the difference scheme (2.22) is unconditionally stable. □

4. Multigrid method for the resulting matrix algebraic equations

We use a V-cycle LOD-MGM to solve the resulting matrix algebraic equations of (1.1). Meanwhile, we show the convergence of the resulting system. In order to develop a fast algorithm, i.e., realizing the computational count $\mathcal{O}(N\log N)$ and the required storage $\mathcal{O}(N)$, as did in [12, 21, 22], we first introduce the Toeplitz matrix and the circulant matrix. The $n \times n$ Toeplitz matrix $T_n(c)$ is defined by [1]

$$T_n(c) := [c_{j-k}]_{j,k=1}^n = \begin{bmatrix} c_0 & c_{-1} & \cdots & c_{-(n-1)} \\ c_1 & c_0 & \cdots & c_{-(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & \cdots & c_0 \end{bmatrix}, \tag{4.1}$$

and the circulant matrices are the "periodic counsins" of Toeplitz matrices. We denote by *circ* $(c_0, c_1, \ldots, c_{n-1})$ the circulant matrix whose first column is $\widetilde{c} = (c_0, c_1, \ldots, c_{n-1})^{\mathrm{T}}$,

$$C_n := \begin{bmatrix} c_0 & c_{n-1} & c_{n-2} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_3 & c_2 \\ c_2 & c_1 & c_0 & \ddots & \ddots & c_3 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & c_{n-3} & \ddots & \ddots & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & c_{n-3} & \cdots & c_1 & c_0 \end{bmatrix}. \tag{4.2}$$

Moreover, we set $\omega_n = \exp(2\pi i/n)$ and put

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{bmatrix},$$

with i as the imaginary unit, and the matrix F_n is called the Fourier matrix. Therefore, a circulant matrix can be diagonalized by the Fourier matrix F_n, i.e.,

$$C_n := F_n^* \mathrm{diag}(F_n \widetilde{c}) F_n, \tag{4.3}$$

where $\mathrm{diag}(F_n \widetilde{c})$ is a diagonal matrix holding the eigenvalues of C_n. From (4.3), we can determine $\mathrm{diag}(F_n \widetilde{c})$ in $\mathcal{O}(N\log N)$ operations by the FFT of the first column $\widetilde{c}$ of C_n [1].

4.1. A $\mathcal{O}(N log N)$ V-cycle MGM for 1D

We employ the V-cycle MGM to solve the one dimensional system (2.11) and illustrate the computational count of $\mathcal{O}(N\log N)$ per iteration and the required storage of $\mathcal{O}(N)$.

Suppose $A_h = I - A^{k+1/2}$, $u_h = U^{k+1}$ and $f_h = (I + A^{k+1/2})U^k + \Delta t F^{k+1/2}$, then the resulting system (2.11) becomes the following general linear system

$$A_h u_h = f_h, \tag{4.4}$$

and the system (4.4) can be carry out by the Algorithm 1 [16, p. 443] and 2.

In Algorithm 1, at the highest (finest grid) level a mesh-size of h is used to solve the resulting system (4.4). The finest grid operator A_h is with the finest grid size $h = \Delta x$; the coarse grid operator $A_H = I - A_{2^l h}$, $H = 2^l h$, for $1 \le l \le \log_2 N - 1$; h_0 is the coarsest mesh-size; I_H^h, I_h^H are respectively the prolongation operator and the restriction operator. For one dimensional system, the restriction operatorI_h^H is defined by [16]

$$I_h^H = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & \cdots & \cdots & \cdots & & & \\ & & & & & 1 & 2 & 1 \end{bmatrix}, \tag{4.5}$$

and the prolongation operator $I_H^h = 2(I_h^H)^{\mathrm{T}}$. The smoothing operator `smooth` may be written as

$$\texttt{smooth}(A_h, u_0, f_h) = S_h u_0 + (I - S_h) A_h^{-1} f_h, \tag{4.6}$$

where S_h is the iteration matrix of the smoothing operator, and we define the weighted (damped) Jacobi iteration matrix by [2, p. 9]

$$S_{h,\omega} = I - \omega D^{-1} A_h, \tag{4.7}$$

with the weighting factor $\omega \in \mathbb{R}$, and D is the diagonal of A_h. Thus, the (4.6) becomes the following weighted Jacobi iteration

$$\texttt{smooth}(A_h, u_0, f_h) = S_{h,\omega} u_0 + \omega D^{-1} f_h. \tag{4.8}$$

In Algorithm 1, the factors ν_1 and ν_2 of $\texttt{smooth}^{\nu_1}(A_h, u_0, f_h)$ and $\texttt{smooth}^{\nu_2}(A_h, u_h, f_h)$ denote the number of weighted Jacobi iterations. In Algorithm 2, we give the stopping criterion of Algorithm 1.

Next, we illustrate the storage requirement of $\mathcal{O}(N)$ and the computational count of $\mathcal{O}(N\log N)$ per iteration.

From (2.12), we have $A^{k+1/2} = \mathrm{diag}(\xi^{k+1/2})\widetilde{A}^{k+1/2}$, where

$$\widetilde{A}^{k+1/2} = \begin{bmatrix} 2g_1^\alpha & g_0^\alpha + g_2^\alpha & g_3^\alpha & \cdots & g_{N_x-2}^\alpha & g_{N_x-1}^\alpha \\ g_0^\alpha + g_2^\alpha & 2g_1^\alpha & g_0^\alpha + g_2^\alpha & g_3^\alpha & \cdots & g_{N_x-2}^\alpha \\ g_3^\alpha & g_0^\alpha + g_2^\alpha & 2g_1^\alpha & g_0^\alpha + g_2^\alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & g_3^\alpha \\ g_{N_x-2}^\alpha & \ddots & \ddots & \ddots & 2g_1^\alpha & g_0^\alpha + g_2^\alpha \\ g_{N_x-1}^\alpha & g_{N_x-2}^\alpha & g_3^\alpha & \cdots & g_0^\alpha + g_2^\alpha & 2g_1^\alpha \end{bmatrix} \tag{4.9}$$

being a Toeplitz matrix, and $\xi^{k+1/2} = [\xi_1^{k+1/2}, \xi_2^{k+1/2} \ldots, \xi_{N_x-1}^{k+1/2}]^T$.

Then, we only need to store $\xi^{k+1/2}$ and $g^\alpha = [2g_1^\alpha, g_0^\alpha + g_2^\alpha, g_3^\alpha \ldots, g_{N_x-1}^\alpha]^T$ which have $2N-2$ parameters, instead of the full matrix $A^{k+1/2}$ which has $(N-1)^2$ parameters, i.e., the required storage $\mathcal{O}(N)$. Consider a one dimensional grid with N points, the finest grid, Ω^h, requires $\mathcal{O}(N)$ storage locations; Ω^{2h} requires 2^{-1} times as much storage as Ω^h; Ω^{4h} requires 4^{-1} times as much storage as Ω^h; in general, Ω^{ph} requires p^{-1} times as much storage as Ω^h. Adding these terms we obtain [2]

$$\text{Storage} = \mathcal{O}(N) \cdot \left(1 + \frac{1}{2} + \frac{1}{2^2} + \ldots, + \frac{1}{2^{\log_2 N - 1}}\right) = \mathcal{O}(N).$$

Taking v a given vector, for the Toeplitz matrix vector multiplication $\widetilde{A}^{k+1/2}v$, we first embed $\widetilde{A}^{k+1/2}$ into a $(2N_x - 2) \times (2N_x - 2)$ circulant matrix, i.e.,

$$\begin{bmatrix} \widetilde{A}^{k+1/2} & * \\ * & \widetilde{A}^{k+1/2} \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} \widetilde{A}^{k+1/2} v \\ \dagger \end{bmatrix}. \tag{4.10}$$

Thus, using (4.3), the computational count of $\widetilde{A}^{k+1/2}v$ remains as $\mathcal{O}(N\log N)$. Therefore, for a V-cycle MGM, each level is visited $\mathcal{O}(N\log N)$ and grid Ω^{ph} requires p^{-1} work units. Similarly, adding these count we have

$$\begin{aligned}&\text{V-cycle MGM computational count}\\ &= \mathcal{O}(N\log N)\cdot\left(1+\frac{1}{2}+\frac{1}{2^2}+\ldots,+\frac{1}{2^{\log_2 N-1}}\right)=\mathcal{O}(N\log N).\end{aligned}$$

By similar analysis, we know that the required storage is still $\mathcal{O}(N)$ and the computational count $\mathcal{O}(N\log N)$ for multidimensional case.

4.2. A $\mathcal{O}(N\log N)$ V-cycle LOD-MGM for 2D

For D-AD scheme (3.3)-(3.4), take

$$\begin{aligned}&A_{h_x}=I-A^{k+1/2}_{2D,\Delta x}; \quad u_{h_x}=U^*; \qquad f_{h_x}=(I+A^{k+1/2}_{2D,\Delta x}+2A^{k+1/2}_{2D,\Delta y})U^k+\Delta t F^{k+1/2};\\ &A_{h_y}=I-A^{k+1/2}_{2D,\Delta y}; \quad u_{h_y}=U^{k+1}; \quad f_{h_y}=U^*-A^{k+1/2}_{2D,\Delta y}U^k.\end{aligned}$$

Similarly, for PR-AD scheme (3.5)-(3.6), denote

$$\begin{aligned}&A_{h_x}=I-A^{k+1/2}_{2D,\Delta x}; \quad u_{h_x}=U^*; \qquad f_{h_x}=(I+A^{k+1/2}_{2D,\Delta y})U^k+\frac{\Delta t}{2}F^{k+1/2};\\ &A_{h_y}=I-A^{k+1/2}_{2D,\Delta y}; \quad u_{h_y}=U^{k+1}; \quad f_{h_y}=(I+A^{k+1/2}_{2D,\Delta x})U^*+\frac{\Delta t}{2}F^{k+1/2}.\end{aligned}$$

Then, both the D-AD and PR-AD schemes, defined by (3.7), reduce to the following LOD form:

$$A_{h_x}u_{h_x}=f_{h_x}, \tag{4.11}$$

$$A_{h_y}u_{h_y}=f_{h_y}. \tag{4.12}$$

Therefore, we can solve the two dimensional system (3.7) by V-cycle LOD-MGM (see Appendix Algorithm 1-3).

The Algorithm 3 starts with the initial time $t=0$ and executes as follows:

(1) First for every fixed $y=y_j$ ($j=1,\ldots,N_y-1$), using Algorithm 1 to solve a set of N_x-1 equations defined by (4.11) at the mesh points $x_i, i=1,\ldots,N_x-1$, to get u_{h_x};

(2) Next alternating the spatial direction, and for each fixed $x=x_i$ ($i=1,\ldots,N_x-1$) solving a set of N_y-1 equations defined by (4.12) at the points $y_j, j=1,\ldots,N_y-1$, once again we employ Algorithm 1 to get u_{h_y}.

4.3. A $\mathcal{O}(NlogN)$ V-cycle LOD MGM for 3D

For D-AD scheme (3.8)-(3.10), similarly, we set

$$\begin{aligned}
&A_{h_x} = I - A^{k+1/2}_{3D,\Delta x}; \quad u_{h_x} = U^{k,1};\\
&f_{h_x} = (I + A^{k+1/2}_{3D,\Delta x} + 2A^{k+1/2}_{3D,\Delta y} + 2A^{k+1/2}_{3D,\Delta z})U^k + \Delta t F^{k+1/2};\\
&A_{h_y} = I - A^{k+1/2}_{3D,\Delta y}; \quad u_{h_y} = U^{k,2}; \quad f_{h_y} = U^{k,1} - A^{k+1/2}_{3D,\Delta y}U^k;\\
&A_{h_z} = I - A^{k+1/2}_{3D,\Delta z}; \quad u_{h_z} = U^{k+1}; \quad f_{h_z} = U^{k,2} - A^{k+1/2}_{3D,\Delta z}U^k;
\end{aligned}$$

Then, the D-AD scheme defined by (3.11), becomes the following form:

$$A_{h_x} u_{h_x} = f_{h_x}, \tag{4.13}$$

$$A_{h_y} u_{h_y} = f_{h_y}, \tag{4.14}$$

$$A_{h_z} u_{h_z} = f_{h_z}. \tag{4.15}$$

Therefore, we can solve the three dimensional system (3.11) by Algorithm 1, 2 and 4.

4.4. Convergence analysis

In this subsection, we discuss the convergence of LOD-MGM. For the convenience of convergence analysis, we assume the coefficient $c(x, y, z, t)$ is constant, and then all $\xi_i^{k+1/2}$ defined above are equal, and it can be denoted as ξ. Let's first consider the one dimensional case.

From (4.9), there exists

$$A_h = I - A^{k+1/2} = I - \xi \widetilde{A}^{k+1/2} \equiv [c_{|j-k|}]_{(N_x-1)\times(N_x-1)}, \tag{4.16}$$

where

$$c_0 = 1 - 2g_1^\alpha \xi; \ c_1 = -(g_0^\alpha + g_2^\alpha)\xi; \ c_k = g_{k+1}^\alpha \xi, \quad k = 2, \ldots, N_x - 2, \tag{4.17}$$

and A_h is a symmetric Toeplitz matrix. From the proof of Theorem 3.2, we know that the matrix A_h is symmetric and strongly diagonally dominant with positive diagonally elements, then A_h is symmetric positive definite [2, p. 3].

Since the matrix A_h is symmetric positive define, we can define the following three different inner products [15, p. 78]

$$(u, v)_0 = (Du, v), \quad (u, v)_1 = (A_h u, v), \quad (u, v)_2 = (D^{-1}A_h u, A_h v), \tag{4.18}$$

where D is the diagonal of A_h and along with their corresponding norms $||\cdot||_i$ $(i = 0, 1, 2)$. If taking the coarsest grid size $H = h_0 = 2h$, Algorithm 1 is called the two-grid method (TGM). The TGM is rarely used in practice since the coarse grid operator may still be too

large to be solved exactly. However, it is useful from a theoretical point view as the first step to study the MGM convergence usually begins from the TGM [12, 15, 16, 23]. Since the MGM convergence analysis is still a challenge topic in computational mathematics [2]. In the following we only consider the convergence of the TGM.

Lemma 4.1. *Let A_h be a M-matrix and the weighted factor $0 < \omega \leq 1$, then weighted Jacobi iteration (4.8) converges.*

Proof. Taking $M = D/\omega$ and $N = D/\omega - A_h$, since $0 < \omega \leq 1$, then M and N be a regular splitting of a matrix A_h. Note that A_h is an M-matrix [2, p. 3], thus we have [16, p. 119]

$$\rho(S_{h,\omega}) = \rho(I - \omega D^{-1} A_h) < 1.$$

□

Lemma 4.2 [15, p. 84]. *Let A_h be a symmetric positive definite and $\eta_0 \geq \rho(D^{-1}A_h)$. If $\sigma \leq \omega(2 - \omega\eta_0)$, then the Jacobi relaxation with relaxation parameter $0 < \omega < 2/\eta_0$ satisfies*

$$||S_{h,\omega} e_h||_1^2 \leq ||e_h||_1^2 - \sigma ||e_h||_2^2, \quad \forall e_h \in \mathbb{R}^{N_x - 1}. \tag{4.19}$$

The inequality (4.19) is called the smoothing condition. For the TGM, the correction operator is given by [2, p. 85]

$$T_c = I - I_H^h (A_H)^{-1} I_h^H A_h,$$

therefore, the convergence factor of the TGM is $||(S_{h,\omega})^{\nu_2} \cdot T_c (S_{h,\omega})^{\nu_1}||_1$; see [15, p. 89]. For convenience, we take $\nu_1 = 0$ and $\nu_2 = 1$. Therefore, the convergence factor of the TGM is given by $||S_{h,\omega} \cdot T_c||_1$.

Lemma 4.3. [15, p. 89] *Let A_h be a symmetric positive definite matrix and $S_{h,\omega}$ satisfy (4.19). Suppose that the interpolation I_H^h has full rank and that, for each e_h,*

$$\min_{e^H \in \mathbb{R}^{N_x/2-1}} ||e_h - I_H^h e_H||_0^2 \leq \kappa ||e_h||_1^2, \quad \forall e_h \in \mathbb{R}^{N_x - 1}, \tag{4.20}$$

with $\kappa > 0$ independent of e_h. Then, $\kappa \geq \sigma$ and the convergence factor of the TGM convergence factor satisfies $||S_{h,\omega} \cdot T_c||_1 \leq \sqrt{1 - \sigma/\kappa}$.

Letting $L_{N_x-1} = \text{tridiag}(-1, 2, -1)$ be the $(N_x - 1) \times (N_x - 1)$ one dimensional discrete laplacian, then L_{N_x-1} is a symmetric positive definite matrix. We define $A_{rest} = A^{k+1/2} + c_1 L_{N_x-1}$, where c_1 is defined by (4.17) and it can also be shown that A_{rest} is symmetric and diagonally dominant with positive diagonally elements; then A_{rest} is positive definite. Hence, we have the following equation

$$(e_h, A_h e_h) = (e_h, (I - c_1 L_{N_x-1} + A_{rest}) e_h) \geq (e_h, (I - c_1 L_{N_x-1}) e_h), \quad \forall e_h \in \mathbb{R}^{N_x - 1}. \tag{4.21}$$

Theorem 4.4. *Since A_h, defined by (4.4), is a symmetric positive definite matrix, if taking $\sigma = \omega(2-\omega\eta_0)$ with $\omega \in (0,1]$, then $S_{h,\omega}$ satisfies (4.19) and the convergence factor of the TGM satisfies*

$$||S_{h,\omega} \cdot T_c||_1 < \sqrt{1-2\sigma/5} < 1, \tag{4.22}$$

where $\eta_0 = \rho(D^{-1}A_h) < 2$.

Proof. From Lemma 4.1, we have $\rho(D^{-1}A_h) < 2$. Taking η_0 and σ in Lemma 4.2 as $\rho(D^{-1}A_h)$ and $\sigma = \omega(2-\omega\eta_0)$, respectively, then $S_{h,\omega}$ satisfies (4.19).

Similar to the proof given in [12], we denote $e_h = (e_1, e_2, \ldots, e_{N_x-1})^{\mathrm{T}} \in \mathbb{R}^{N_x-1}$, $e_0 = e_{N_x} = 0$, and $e_H = (e_2, e_4, \ldots, e_{N_x-2})^{\mathrm{T}} \in \mathbb{R}^{N_x/2-1}$; the norm $||\cdot||_0$ is defined by (4.18), and $D = \mathrm{diag}(A_h) = c_0 I$, where c_0 is defined by (4.17). There exists

$$||e_h - I_H^h e_H||_0^2 = c_0 \sum_{i=0}^{N_x/2-1}\left(e_{2i+1} - \frac{1}{2}e_{2i} - \frac{1}{2}e_{2i+2}\right)^2 \le c_0 \sum_{i=1}^{N_x-1}\left(e_i^2 - e_i e_{i+1}\right).$$

From the above inequality, we obtain

$$\sum_{i=1}^{N_x-1} e_i^2 \ge \sum_{i=1}^{N_x-1} e_i e_{i+1}.$$

Similarly, we can check that

$$\sum_{i=1}^{N_x-1} e_i^2 \ge -\sum_{i=1}^{N_x-1} e_i e_{i+1}. \tag{4.23}$$

Combining (4.21) with (4.23), we have

$$\begin{aligned} ||e_h||_1^2 = (e_h, A_h e_h) \ge (e_h, (I - c_1 L_{N_x-1})e_h) = \sum_{i=1}^{N_x-1}\big((1-2c_1)e_i^2 + 2c_1 e_i e_{i+1}\big) \\ = \sum_{i=1}^{N_x-1}\left(\left(\frac{1}{2} - 2c_1\right)(e_i^2 - e_i e_{i+1}) + \frac{1}{2}(e_i^2 + e_i e_{i+1})\right) \ge \left(\frac{1}{2} - 2c_1\right)\sum_{i=1}^{N_x-1}\left((e_i^2 - e_i e_{i+1})\right). \end{aligned}$$

Hence, Eq. (4.20) holds, i.e.,

$$||e_h - I_H^h e_H||_0^2 \le \kappa ||e_h||_1^2,$$

where $$\kappa = \frac{c_0}{\frac{1}{2} - 2c_1} = \frac{2c_0}{1-4c_1} = \frac{2 + 4\xi(-g_1^\alpha)}{1 + 4\xi(g_0^\alpha + g_2^\alpha)} \in \left(1, \frac{5}{2}\right),$$

since, according to Lemma 3.1, it is easy to check that $g_0^\alpha + g_2^\alpha < -g_1^\alpha < \frac{5}{2}(g_0^\alpha + g_2^\alpha)$. From Lemma 4.3, we have $||S_{h,\omega} \cdot T_c||_1 < \sqrt{1-2\sigma/5} < 1$. □

Theorem 4.4 shows that the V-cycle scheme has a convergence factor with bound, that is independent of Δx.

For the multidimensional case, we can similarly define the weighted Jacobi iteration matrix

$$S_{h_x,\omega} = I - \omega D_{h_x}^{-1} A_{h_x}, \quad S_{h_y,\omega} = I - \omega D_{h_y}^{-1} A_{h_y}, \quad S_{h_z,\omega} = I - \omega D_{h_z}^{-1} A_{h_z}$$

and the LOD-TGM correction operators

$$T_{c_x} = I - I_H^h (A_H)^{-1} I_h^H A_{h_x}, \quad T_{c_y} = I - I_H^h (A_H)^{-1} I_h^H A_{h_y}, \quad T_{c_z} = I - I_H^h (A_H)^{-1} I_h^H A_{h_z},$$

then the convergent factors of the LOD-TGM satisfy $||S_{h_x,\omega} \cdot T_{c_x}||_1 < \sqrt{1-2\sigma/5} < 1$, $||S_{h_y,\omega} \cdot T_{c_y}||_1 < \sqrt{1-2\sigma/5} < 1$ and $||S_{h_z,\omega} \cdot T_{c_z}||_1 < \sqrt{1-2\sigma/5} < 1$.

Remark 4.7 As mentioned in the Introduction section, nowadays there are two different second-order discretization schemes for fractional operators [17, 20]; all the analysis given in this paper can be parallel extended to the case that the fractional operators are discretized by the scheme given in [20], in fact the scheme given in [20] has more wide applications because of its good properties; and in Table 4, we show the numerical results obtained by using the scheme of [20] to discretize the fractional operators of (2.13).

5. Numerical results

We employ the V-cycle MGM and V-cycle LOD-MGM described in Section 4 to solve the one dimensional case (2.8) and multidimensional case (2.13,1.1), respectively. The stopping criterion is taken as

$$\frac{||r^{(l)}||_2}{||r^{(0)}||_2} < 10^{-7},$$

where $r^{(l)}$ is the residual vector after l iterations. In all tables, N_t denotes the number of time steps; N_x, N_y and N_z, respectively, denotes the number of spatial grid points in x, y, and z direction, and the numerical errors are measured by the l_∞ norm, 'Rate' denotes the convergent orders. 'CPU' denotes the total CPU time in seconds (s) or minutes (m) for solving the resulting discretized systems, and 'Iter' denotes the average number of iterations required to solve a general linear system $A_h u_h = f_h$ at each time level.

All numerical experiments are programmed in Python, and each computation was carried out on a PC with the configuration: AMD Phenom (tm) II X4 830 CPU 2.79 GHZ and 3 GB RAM and a Linux operating system. All the numerical results listed in the following tables are got by the V-cycle MGM or V-cycle LOD-MGM with the parameters: the number of iterations $(\nu^1, \nu^2) = (1, 1)$ and $(\omega_{pre}, \omega_{post}) = (1, 1/2)$.

Remark 5.1. From our numerical experiences, we find that:

(1) With the increasing of the order of fractional derivative α from 1 to 2, the condition number of the matrix $\widetilde{A}^{k+1/2}$ becomes bigger and bigger; and when getting the same accuracy the cost for $\alpha = 1.9$ almost double the cost for $\alpha = 1.1$;

(2) For making MGM more efficient, the parameters can be dynamically chosen as when α increases from 1 to 2, correspondingly ω_{post} decreases from 1 to 0.5 and fixing $\omega_{pre} = 1$ or fixing $\omega_{post} = 1$ and ω_{pre} decreases from 1 to 0.5;

(3) MGM is still powerful for second-order schemes, when simulating (2.8) with the parameters given in subsection 5.1, in Table 1 it is shown that the second-order scheme costs 4.82 s to obtain the accuracy with the maximum error $1.2407e-006$, and the first-order scheme used in [12] costs 413.95 s when getting the accuracy with the maximum error $2.0358e-006$.

5.1. Numerical results for 1D

Let us consider the one dimensional Riesz fractional diffusion equation (2.8), where $0 < x < 1$ and $0 < t \leq 1$, with the variable coefficient $c(x,t) = x^\alpha t$, the forcing function

$$\begin{aligned} f(x,t) &= -e^{-t}x^2(1-x)^2 \\ &\quad + \frac{x^\alpha t e^{-t}}{cos(\alpha\pi/2)}\left[\frac{x^{2-\alpha}+(1-x)^{2-\alpha}}{\Gamma(3-\alpha)} - 6\frac{x^{3-\alpha}+(1-x)^{3-\alpha}}{\Gamma(4-\alpha)} + 12\frac{x^{4-\alpha}+(1-x)^{4-\alpha}}{\Gamma(5-\alpha)}\right], \end{aligned}$$

and the initial condition $u(x,0) = x^2(1-x)^2$ and the boundary conditions $u(0,t) = u(1,t) = 0$. This fractional PDE has the exact value $u(x,t) = e^{-t}x^2(1-x)^2$, which may be confirmed by applying the fractional differential equations

$$\begin{aligned} {}_{x_L}D_x^\nu (x-x_L)^p &= \frac{\Gamma(p+1)}{\Gamma(p+1-\nu)}(x-x_L)^{p-\nu}, \\ {}_xD_{x_R}^\nu (x_R-x)^p &= \frac{\Gamma(p+1)}{\Gamma(p+1-\nu)}(x_R-x)^{p-\nu}. \end{aligned}$$

From Table 1, we numerically confirm that the numerical scheme has second-order accuracy and the computational cost is of $\mathcal{O}(N\log N)$ operations.

5.2. Numerical results for 2D

Consider the two dimensional Riesz fractional convection diffusion equation (2.13), on a finite domain $0 < x < 1$, $0 < y < 1$, $0 < t \leq 1$, and with the variable coefficients

$$c(x,y,t) = x^\alpha y, \quad d(x,y,t) = xy^\beta,$$

and the initial condition $u(x,y,0) = x^2(1-x)^2y^2(1-y)^2$ and the Dirichlet boundary conditions on the rectangle in the form $u(0,y,t) = u(x,0,t) = 0$ and $u(1,y,t) = u(x,1,t) = 0$

Table 1: MGM to solve the resulting system (2.11) of the 1D Riesz fractional convection diffusion equation (2.8) at $t = 1$ and $N_t = N_x$.

N_t, N_x	$\alpha = 1.1$	Rate	Iter	CPU	$\alpha = 1.9$	Rate	Iter	CPU
2^5	7.8755e-005		4.0	0.19 s	7.5578e-005		6.0	0.28 s
2^6	2.1801e-005	1.8530	4.0	0.45 s	1.9255e-005	1.9727	6.0	0.74 s
2^7	5.6999e-006	1.9354	4.0	1.29 s	4.8923e-006	1.9766	6.0	2.14 s
2^8	1.4565e-006	1.9684	3.0	2.56 s	1.2407e-006	1.9794	6.0	4.82 s
2^9	3.8540e-007	1.9181	3.0	7.45 s	3.1420e-007	1.9814	6.0	13.56 s
2^{10}	9.7292e-008	1.9860	3.0	18.63 s	8.1028e-008	1.9552	6.0	34.99 s

for all $t \geq 0$. The exact solution to this two dimensional Riesz fractional convection diffusion equation is

$$u(x, y, t) = e^{-t}x^2(1-x)^2y^2(1-y)^2.$$

From the above given quantities, it is easy to obtain the forcing function $f(x, y, t)$.

Table 2: LOD-MGM to solve the resulting system of the 2D Riesz fractional convection diffusion equation (2.13) by the D-AD scheme (2.17)-(2.18) at $t = 1$ and $N_t = N_x = N_y$.

N_t, N_x, N_y	$\alpha = 1.1, \beta = 1.1$	Rate	Iter	CPU	$\alpha = 1.8, \beta = 1.9$	Rate	Iter	CPU
2^4	2.4698e-005		4.5	2.09 s	2.5475e-005		7.0	3.51 s
2^5	6.1249e-006	2.0117	4.0	11.26 s	6.5211e-006	1.9659	6.0	18.03 s
2^6	1.5212e-006	2.0095	4.0	55.85 s	1.6662e-006	1.9686	6.0	90.73 s
2^7	3.7812e-007	2.0083	4.0	5 m 17 s	4.2362e-007	1.9757	6.0	8 m 45 s
2^8	9.4076e-008	2.0070	3.0	21 m 36 s	1.0744e-007	1.9792	6.0	39 m 1 s

From Table 2 and Table 3, numerically it can also be noticed that the D-AD and PR-AD are equivalent in two dimensional problems. We employ the LOD-MGM to solve two dimensional Riesz fractional diffusion equation, numerical results further display the computational cost is of $\mathcal{O}(N\log N)$ operations and the numerical scheme is second-order convergent.

In particular, we further numerically confirm that this paper is still valid if the Riesz fractional derivative (1.2) is discretized by another existing second-order discretization scheme given in [20], see Table 4. In fact, theoretically we can also easily draw the same conclusion.

Table 3: LOD-MGM to solve the resulting system of the 2D Riesz fractional convection diffusion equation (2.13) by the PR-AD scheme (2.19)-(2.20) at $t = 1$ and $N_t = N_x = N_y$.

N_t, N_x, N_y	$\alpha = 1.1, \beta = 1.1$	Rate	Iter	CPU	$\alpha = 1.8, \beta = 1.9$	Rate	Iter	CPU
2^4	2.4698e-005		4.5	2.05 s	2.5475e-005		7.0	3.48 s
2^5	6.1249e-006	2.0117	4.0	11.17 s	6.5211e-006	1.9659	6.0	17.71 s
2^6	1.5212e-006	2.0095	4.0	55.14 s	1.6662e-006	1.9686	6.0	90.24 s
2^7	3.7812e-007	2.0083	4.0	5 m 18 s	4.2362e-007	1.9757	6.0	8 m 42 s
2^8	9.4075e-008	2.0070	4.0	22 m 25 s	1.0744e-007	1.9792	6.0	39 m 31 s

Table 4: LOD-MGM to solve the resulting system of the 2D Riesz fractional convection diffusion equation (2.13) by the D-AD scheme (2.17)-(2.18) at $t = 1$ and $N_t = N_x = N_y$; **the Riesz fractional derivative (1.2) is discretized by the scheme given in [20]**.

N_t, N_x, N_y	$\alpha = 1.1, \beta = 1.1$	Rate	Iter	CPU	$\alpha = 1.8, \beta = 1.9$	Rate	Iter	CPU
2^4	2.4592e-005		5.0	2.25 s	2.4532e-005		7.0	3.5 s
2^5	6.1745e-006	1.9938	4.0	11.42 s	6.0897e-006	2.0102	6.0	17.98 s
2^6	1.5426e-006	2.0010	4.0	56.57 s	1.5102e-006	2.0116	6.0	90.74 s
2^7	3.8444e-007	2.0045	4.0	5 m 28 s	3.7350e-007	2.0155	6.0	8 m 41 s
2^8	9.5743e-008	2.0055	4.0	22 m 43 s	9.2374e-008	2.0155	6.0	38 m 22 s

5.3. Numerical results for 3D

Consider the three dimensional Riesz fractional convection diffusion equation (1.1), on a finite domain $0 < x < 1$, $0 < y < 1$, $0 < z < 1$, and $0 < t \leq 1$, and with the variable coefficients

$$c(x, y, z, t) = x^{\alpha} yz, \quad d(x, y, z, t) = xy^{\beta} z, \quad e(x, y, z, t) = xyz^{\gamma},$$

and the initial condition $u(x, y, z, 0) = x^2(1-x)^2 y^2 (1-y)^2 z^2 (1-z)^2$ and the zero Dirichlet boundary conditions on the cube. The exact solution to this three dimensional Riesz fractional convection diffusion equation is

$$u(x, y, z, t) = e^{-t} x^2 (1 - x)^2 y^2 (1 - y)^2 z^2 (1 - z)^2.$$

According to the above conditions, it is easy to obtain the forcing function $f(x, y, z, t)$.

The numerical results in Table 5 are obtained by employing the LOD-MGM to solve the three dimensional Riesz fractional diffusion equation, they again display the computational count of $\mathcal{O}(N\log N)$ operations and the scheme is second-order convergent.

Table 5: LOD-MGM to solve the scheme (2.23)-(2.25) of the 3D Riesz fractional convection diffusion equation (1.1) at $t = 1$ and $N_t = N_x = N_y = N_z$.

N_t	$\alpha=\beta=\gamma=1.1$	Rate	Iter	CPU	$\alpha=1.8,\beta=1.9,\gamma=1.8$	Rate	Iter	CPU
2^3	5.9349e-006		4.75	3.31 s	5.8311e-006		7.0	5.28 s
2^4	1.4792e-006	2.0044	4.0	42.19 s	1.4867e-006	1.9717	7.0	70.1 s
2^5	3.7377e-007	1.9846	4.5	7 m 40 s	3.8428e-007	1.9519	6.0	13 m 1 s
2^6	9.3376e-008	2.0010	4.37	76 m 13 s	9.8179e-008	1.9687	6.0	136 m 1 s

6. Conclusions

With the appearing of the two effective second-order discretization schemes [17, 20] and MGM being successfully employ to solve the resulting system of the one dimensional fractional diffusion equation discretized by first-order scheme [12], our attentions turn to the possibility of efficiently solving the multidimensional fractional Riesz diffusion equation by second-order scheme and MGM. This paper shows that when solving (1.1) the second-order schemes are unconditionally stable, and the structure of the resulting matrix algebraic equations is almost the same as the one by the first-order scheme and then none computational costs increase but accuracy is greatly improved if using the second scheme instead of the first-order one. And LOD-MGM still preserves its powerfulness of $\mathcal{O}(N\log N)$ computational counts and of $\mathcal{O}(N)$ storage when solving the resulting matrix system of the multidimensional fractional Riesz diffusion equation discretized by the second-order scheme. For computing the completely same equation, from Table 1, it can be noticed that the second-order scheme costs $4.82\,s$ to obtain the accuracy with the maximum error $1.2407e-006$, and the first-order scheme used in [12] costs $413.95\,s$ when getting the accuracy with the maximum error $2.0358e-006$.

Last but not least, we want to refer to that although this paper focus on using the second-order discretization given in [17], all the analysis of this paper is still valid if applying the second-order discretization in [20]; in fact the validness is already verified numerically in Table 4.

Acknowledgments

This work was supported by the Program for New Century Excellent Talents in University under Grant No. NCET-09-0438, the National Natural Science Foundation of China under Grant No. 10801067 and No. 11271173, and the Fundamental Research Funds for the Central Universities under Grant No. lzujbky-2010-63 and No. lzujbky-2012-k26.

Appendix

For a general linear system

$$A_h u_h = f_h,$$

we employ the following V-cycle MGM (Algorithm 1-2) to solve one dimensional (2.8) and V-cycle LOD-MGM (Algorithm 1-3) solve two dimensional (2.13). Solve the three-dimensional system (1.1) by Algorithm 1,2 and 4.

Algorithm 1 MGM for 1D u_h=V-cycle(A_h, u_0, f_h)

```
Pre-smooth:   u_h := smooth^{ν1}(A_h, u_0, f_h)
Get residual:   r_h = f_h − A_h u_h
Coarsen:   r_H = I_h^H r_h
if H == h_0 then
    Solve:   A_H ξ_H = r_H
else
    Recursion:   ξ_H = V-cycle(A_H, 0, r_H)
end if
Correct:   u_h := u_h + I_H^h ξ_H
Post-smooth:   u_h := smooth^{ν2}(A_h, u_h, f_h)
Return u_h
```

Algorithm 2 Stopping criterion for MGM

```
u_h := u_0
r0 := ||A_h u_0 − f_h||_2
r_h := r_0
while r_h / r_0 > ϵ do
    u_h := V-cycle(A_h, u_h, f_H)
    r_h := ||A_h u_h − f_h||_2
end while
Return u_h
```

References

[1] A. Böttcher, S.M. Grudsky, Spectral properties of banded Toeplitz matrices, SIAM, Philadelphia, PA, 2005.

Algorithm 3 LOD-MGM for 2D

```
t := 0
while t < T do
    t := t + Δt
    for every fixed y_j, (j = 1 : N_y − 1) do
        solve system (4.11) by Algorithm 2
    end for
    for each fixed x_i, (i = 1 : N_x − 1) do
        solve system (4.12) by Algorithm 2
    end for
end while
```

[2] W.L. Briggs, V.E. Henson, S.F. Mccormick, A Multigrid Tutorial, Second Edition, SIAM, Philadelphia, 2000.

[3] M.H. Chen, W.H. Deng, Y.J. Wu, Second order finite difference approximations for the two-dimensional time-space Caputo-Riesz fractional diffusion equation, arXiv:1207.2012v1 [math.NA].

[4] J. Dougls, On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods, J. Soc. Indust. Appl. Math., 3 (1955) 42-65.

[5] J. Dougls, Alternating direction methods for three space variables, Numer. Math., 6 (1964) 428-453.

[6] J. Dougls, S. Kim, Improved accuracy for locally one-dimensional Methods for parabolic equations, Math. Mod. Meth. Appl. Sci., 11 (9) (2001) 1563-1579.

[7] J. Dougls, D. Peaceman, Numerical solution of two-dimensioal heat flow problems, American Institute of Chemical Engineering Journal, 1 (1955) 505-512.

[8] E. Isaacson, H.B. Keller, Analysis of Numerical Methods, Wiley, New York, 1966.

[9] F. Mainardi, Fractional calculus and waves in linear viscoelasticity, Imperial College Press, London, 2010.

[10] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math., 56 (2006) 80-90.

[11] K.S. Miller, B. Ross, An Introduction to the Fractional Calculus and Fractional Differential Equations, Wiley-Interscience Publication, US, 1993.

Algorithm 4 LOD-MGM for 3D

```
t := 0
while t < T do
    t := t + Δt
    for every fixed z_l, (l = 1 : N_z − 1) and y_j, (j = 1 : N_y − 1) do
        solve system (4.13) by Algorithm 2
    end for
    for each fixed x_i, (i = 1 : N_x − 1) and z_l, (l = 1 : N_z − 1) do
        solve system (4.14) by Algorithm 2
    end for
    for each fixed y_j, (j = 1 : N_y − 1) and x_i, (i = 1 : N_x − 1) do
        solve system (4.15) by Algorithm 2
    end for
end while
```


[12] H. Pang, H. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys., 231 (2012) 693-703.

[13] D. Peaceman, H. Rachford, The numerical solution of parabolic and elliptic differential differential equaitons, J. Soc. Indust. Appl. Math., 3 (1955) 28-41.

[14] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.

[15] J. Ruge, K. Stuben, Algebraic Multigrid, in: S. McCormick (Ed.), Multigrid Methods, vol. 4, SIAM, Philadelphia, 1987.

[16] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.

[17] E. Sousa, C. Li, A weighted finite difference method for the fractional diffusion equation based on the Riemann-Liouville derivative, arXiv:1109.2345v1 [math.NA].

[18] C. Tadjeran, M.M. Meerschaert, A second-order accurate numerical method for the two-dimensional fractional diffusion equation, J. Comput. Phys., 220 (2007) 813-823.

[19] V.E. Tarasov, Fractional Dynamics: Applications of fractional calculus to dynamics of particles, fields and media, Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg 2010.

[20] W.Y. Tian, H. Zhou, W.H. Deng, A Class of Second Order Difference Approximations for Solving Space Fractional Diffusion Equations, arXiv:1201.5949v3 [math.NA].

[21] H. Wang, K. Wang, T. Sircar, A direct $\mathcal{O}(N\log^2 N)$ finite difference method for fractional diffusion equations, J. Comput. Phys., 229 (2010) 8095-8014.

[22] H. Wang, T. Basu, A fast finite difference method for two-dimensional space-fractional diffusion equations, SIAM J. Sci. Comput., 34 (2012) A2444-A2458.

[23] P. Wesseling, An Introduction to Multigrid Methods, John Wiley & Sons Ltd., Chichester, 1992.

[24] Q. Yang, F. Liu, I. Turner, Numerical methods for fractional partial differential equations with Riesz space fractional derivatives, Appl. Math. Model., 34 (2010) 200-218.

[25] H. Zhou, W.Y. Tian, W.H. Deng, Quasi-compact finite difference schemes for space fractional diffusion equations, J. Sci. Comput., doi: 10.1007/s10915-012-9661-0.

[26] P. Zhuang, F. Liu, V. Anh and I. Turner, Numerical methods for the variable-order fractional advection-diffusion equation with a nonlinear source term, SIAM J. Numer. Anal., 47 (2009) 1760-1781.